Camera Integration DLL for Pleora iPort SDK

User manual
Version 1.1.0

Art. Nr.: TBD

Date: April 2010

Technical details are subject to change

VDS Vosskühler GmbH
Weiße Breite 7 · D-49084 Osnabrück
Telefon +49 / (0)541 / 80084-0 · Telefax +49 / (0)541 / 80084-10
http://www.vdsvossk.de · email: vds@vdsvossk.de

VDS Vosskühler

Contents

1 Introduction

The „Camera Integration DLL" can be thought of as a driver that contains code to easiely set up the camera's parameters. It is designed to fit seamlessly into Pleora's software framework and to provide VDS Vosskühler specific camera functionality to software developers. It can be used together with the VDS Vosskühler GIP-1000 Interface, the Pleora GigE adapter "PT1000-CL2" or compatible.
The Camera Integration DLL supports nearly all cameras of VDS Vosskühler. However, not all cameras of VDS Vosskühler are tested together with the Pleora adapter.

1.1 When do I need this product?

All software that uses the Pleora iPort SDK and that makes use of the internal camera management framework can benefit from this DLL. As an example, the "Coyote" software tool included in Pleora's SDK does so.
Applications from VDS Vosskühler such as AcquireControl or HCC1000Control do not need this DLL since they access the Pleora iPort SDK on a lower level.
If you are a software developer, and if you are planning to integrate a VDS Vosskühler camera based on the GIP-1000 module into your software, you are encouraged to use the Camera Integration DLL since it simplifies controlling camera specific functions. Nevertheless using this DLL is not mandatory.

1.2 What is new?

- Starting with version 1.1.0, the camera integration library also supports the 64 bit versions of the Pleora iPort SDK.
- The Debug versions of the Camera Integration DLL and the .LIB files that were present in versions prior to 1.1.0, have been removed, since they were not required.

2 Quickstart

There are two options installing this product on your system. In most cases, the automatic installation will be the best way. If you prefer manual installation or in case the automatic installation does not work, you may choose the manual installation as an alternative.

2.1 Automatic installation

√ Install the Pleora iPort SDK on your PC first before installing the camera integration DLL. You can find the SDK on VDS Vosskühler’s product support CD. Alternatively, you can download the latest version from VDS Vosskühler’s Web site. The installer’s executable is usually named “IPEngineSDK.exe”, “eBUS-Vision Package.exe” or similar.

√ Get VDS Vosskühler’s Camera integration DLL package. You can find it on VDS Vosskühler’s product support CD. Alternatively, you can download the latest version from VDS Vosskühler’s Web site. When downloading from the internet, you will receive a packed Archive in “ZIP”-format. It is usually named “PleoraCamInt_VXXX.zip” (where XXX denotes the version of the Camera Integration DLL). You might need a 3rd-party software that is capable of decompressing ZIP-Archives, such as “WinZip”, “WinRar” or “7-Zip”. Decompress the archive to an empty folder on your hard drive.

√ Execute the „install.cmd“-script in the root of the archive tree. It will copy the necessary files into the Pleora-SDK tree, so that they will be found by the framework. **Note that you must have administrative rights to successfully execute the script!** On Windows Vista and on Windows 7, you might right click onto the “install.cmd” inside the Explorer and choose “Execute as administrator” from the context menue.

2.2 Manual Installation

√ Install the Pleora iPort SDK on your PC first before installing the camera integration DLL. You can find the SDK on VDS Vosskühler’s product support CD. Alternatively, you can download the latest version from VDS Vosskühler’s Web site. The installer’s executable is usually named “IPEngineSDK.exe”, “eBUS-Vision Package.exe” or similar.

√ Get VDS Vosskühler’s Camera integration DLL package. You can find it on VDS Vosskühler’s product support CD. Alternatively, you can download the latest version from VDS Vosskühler’s Web site. When downloading from the internet, you will receive a packed Archive in “ZIP”-format. It is usually named “PleoraCamInt_VXXX.zip” (where XXX denotes the version of the Camera Integration DLL). You might need a 3rd-party software that is capable of decompressing ZIP-Archives, such as “WinZip”, “WinRar” or “7-Zip”. Decompress the archive to an empty folder on your hard drive.

√ Now, the following files must be copied into the tree of the Pleora iPort SDK. The following table shows all source files relative to the root directory of the ZIP-archive in the left column. The destination directory depends on where you have installed the Pleora iPort SDK and which version of the SDK you are using.

SDK's with version number < 3.0 are installed to “C:\Program Files\Pleora Technologies Inc.\iPORT Software” by default. The SDK-Setup creates an environment variable named “PLEORA_ROOT”. This variable contains the destination directory. If you are not sure, open a command shell and enter “echo %PLEORA_ROOT%”. This will display the content of the variable and show you, where to copy the files.

Source file	Destination directory	Description
Binaries\CyCamVosskuehler_SDK_V22X.dll, Binaries\CyCamVosskuehler_SDK_V23X.dll	%PLEORA_ROOT%\Binaries	The camera integration DLLs. It is sufficient to copy only the files that are appropriate for your SDK-version.
Includes\CyVosskuehlerCamera.h	%PLEORA_ROOT%\Includes	Header file that contains definitions for the additional parameters provided by the camera-interface
Includes\CyVosskuehlerPixelConverters.h	%PLEORA_ROOT%\Includes	Header file that defines additional pixel converters

Table 1 – Installation locations for files using Pleora SDK's < V3.X

Starting with SDK version 3.0, Pleora has completely changed the installation locations of files. The "PLEORA_ROOT" variable has been replaced by "IPORT_ROOT". The integration DLL's files must be placed into the "Pleora" subdirectory in your Common Program Files directory. If you want to find out the current settings of the environment variables, open a command shell and enter "echo %IPORT_ROOT%" and "echo %CommonProgramFiles%". This will display the content of the variables and show you, where to copy the files.

Source file	Destination directory	Description
On 32 bit systems: Binaries\CyCamVosskuehler_SDK_V23X.dll, Binaries\CyCamVosskuehler_SDK_V31X.dll, Binaries\msvcp100.dll, Binaries\msvcr100.dll **On 64 bit Systems:** Binaries64\CyCamVosskuehler_SDK_V31_64.dll, Binaries64\msvcp100.dll, Binaries64\msvcr100.dll	%CommonProgramFiles%\Pleora	The camera integration DLLs. It is sufficient to copy only the files that are appropriate for your SDK-version. The msvc?100.dll files are only required if they are not yet installed on your system.
Includes\CyVosskuehlerCamera.h	%IPORT_ROOT%\Includes	Header file that contains definitions for the additional parameters provided by the camera-interface
Includes\CyVosskuehlerPixelConverters.h	%IPORT_ROOT%\Includes	Header file that defines additional pixel converters

Table 2 – Installation location for files using Pleora SDK's >= V3.X

2.3 Test the Camera Integration DLL

Now that all files have been copied into the appropriate directories, the Camera Integration DLL may be tested using Pleora's Coyote software that is part of the SDK. Connect the camera to the network, and click on the “detect“ button. Assign an IP-address to the camera and select it. Now, when clicking on “Select Camera“, the entry “VDS Vosskuehler GmbH“ should occur in the selection list:

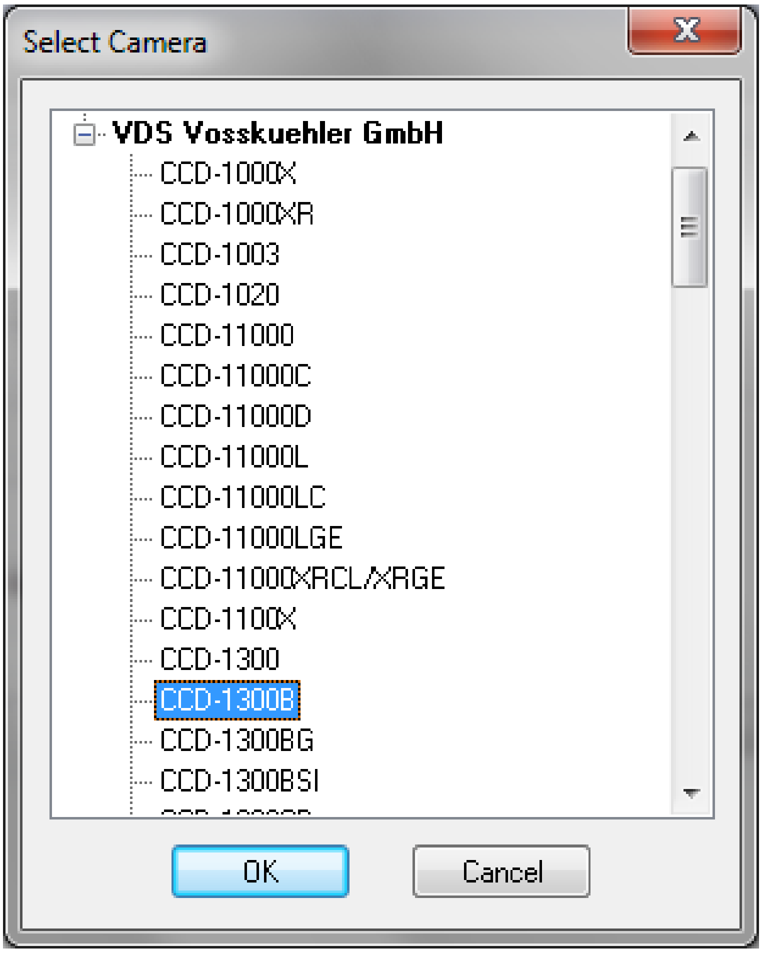
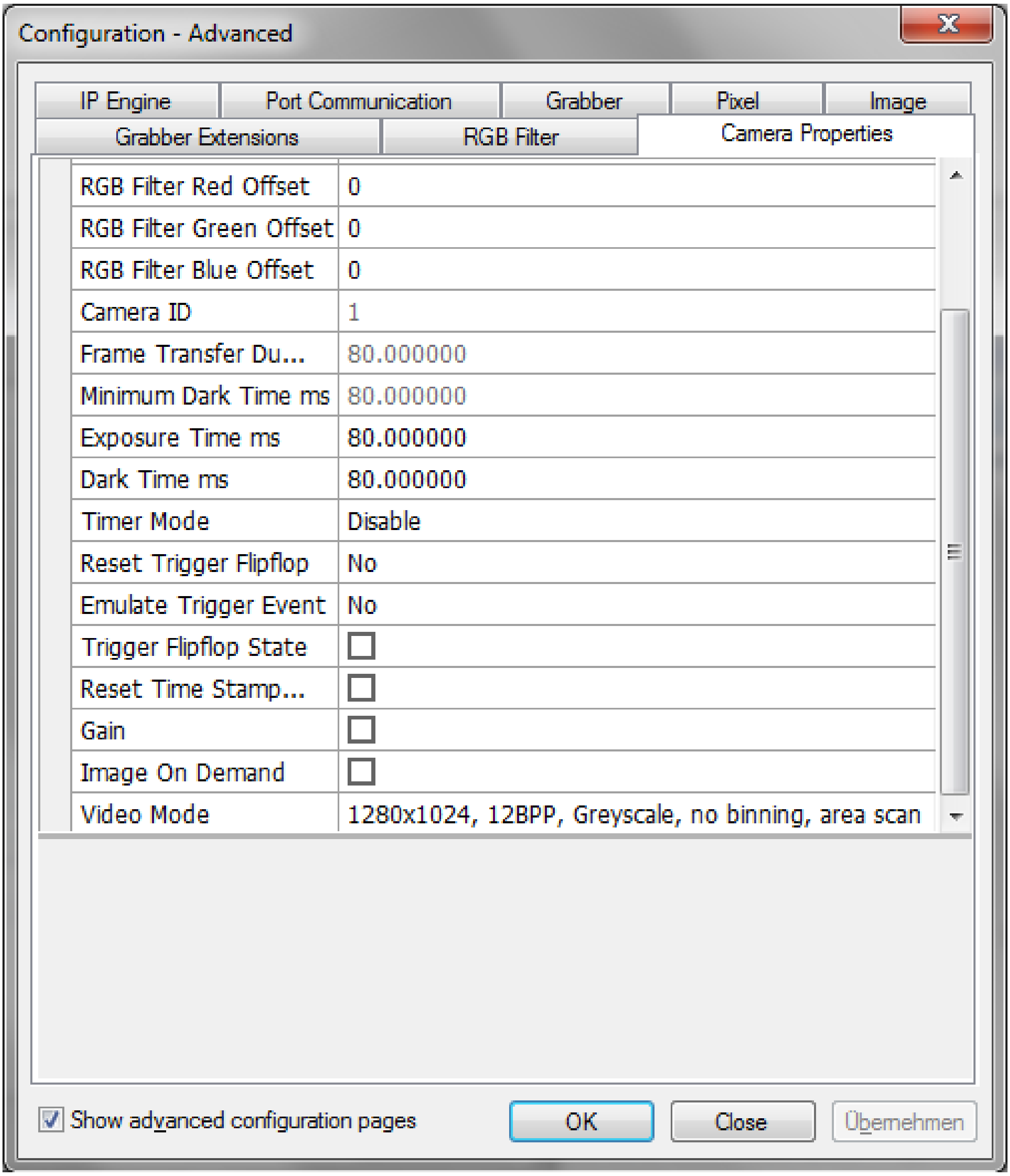


Figure 1 - Select the camera

Select the appropriate camera type and click „Ok“.

Now click the “configure“-button and select the “Camera Properties“ sheet. The camera's features can be controlled from here:



RGB Filter Red Offset	0
RGB Filter Green Offset	0
RGB Filter Blue Offset	0
Camera ID	1
Frame Transfer Du...	80.000000
Minimum Dark Time ms	80.000000
Exposure Time ms	80.000000
Dark Time ms	80.000000
Timer Mode	Disable
Reset Trigger Flipflop	No
Emulate Trigger Event	No
Trigger Flipflop State	□
Reset Time Stamp...	□
Gain	□
Image On Demand	□
Video Mode	1280x1024, 12BPP, Greyscale, no binning, area scan

☑ Show advanced configuration pages

Figure 2 - Camera properties (example)

Changes will be applied directly after clicking the “Apply“-button. Note that the entries may look different when selecting other cameras than “CCD-1300B” as in this example.

3 VDS Vosskühler specific parameters

The Camera Integration DLL implements parameters that provide easy access to camera specific features. This chapter explains all parameters that are defined by the Camera Integration DLL. They can be easily checked out using Pleora's Coyote application by clicking the “Configure”-button and opening the “Camera Properties” sheet. Every parameter discussed in this chapter has a representation in the “CyCameraVosskuehler.h” header file. Controlling parameters programmatically in your own software is described in the next chapter.

3.1 Camera ID

The camera ID parameter is a 32 bit value, that is defined by VDS Vosskühler to identify a specific camera model. This parameter is read only.

3.2 Frame Transfer Duration ms (t_{FTD})

The frame transfer duration parameter specifies the amount of time in milliseconds, that the camera requires to output an acquired image to the GIP-1000 module. This value is required to calculate exposure time and dark time limitations. Note that this value is dependent on the video mode and may change if the video mode changes. This parameter is read only.

3.3 Minimum Dark Time ms (t_{MDT})

The minimum dark time parameter specifies the minimum amount of time that must be kept between two exposure cycles in milliseconds. This parameter is read only.

3.4 Exposure Time ms ($t_{\exp}$)[1]

The exposure time parameter specifies the amount of time the camera sensor is exposed to light, that is the integration time in milliseconds. Note that the exposure time is usually rounded to a multiple of the duration of one camera line. You can obtain the length of a line from the camera’s manual. Note that this parameter has only effect in the following timer modes: “Continuous”, “Trigger Once”, “Start By Trigger”, “Stop By Trigger” and “Start Stop By Trigger”.

[1] VDS Vosskühler also uses the term “Shutter Time” as a synonym for “Exposure Time”.

3.5 Dark Time ms (t_{Dark})[2]

The dark time parameter specifies the time between two exposure cycles in milliseconds. Note that this time must be chosen to meet camera specific limitations. For most VDS Vosskühler cameras, the following rules apply:

$t_{Dark} \geq t_{MDT}$ (This conditions must always be met.)

if $t_{MDT} < t_{FTD}$ then

> The camera is multi trigger capable. That means, that a new exposure cycle may be started, while the previous image is still read out of the camera. The condition $(t_{\exp} + t_{Dark}) \geq t_{FTD}$ must be met.

else

> The camera is not multi trigger capable. That means, that a new exposure cycle must not be started, until image readout has finished. The condition $t_{Dark} \geq t_{FTD}$ must be met.

Please always refer to the camera's manual. Some cameras may have exceptional rules.

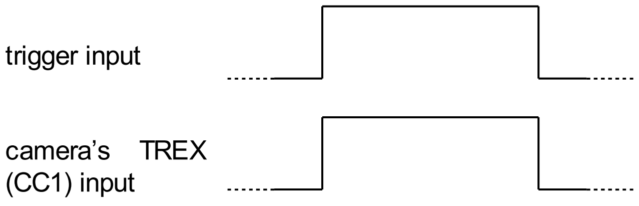
Note that this parameter has only effect in the following timer modes: “Continuous”, “Trigger Once”, “Start By Trigger”, “Stop By Trigger” and “Start Stop By Trigger”.

3.6 Timer Mode

The timer mode parameter controls the way how the camera is exposing images. The Camera Integration DLL implements an exposure control unit, that allows multiple ways of triggering the exposure. Figure 3 shows the basic structure of this unit. When using the VDS Vosskühler GIP-1000 module, the trigger input can be found on pin 10 and 11 of the 15 pin SUB-D connector. In the GIP-1000 module, the trigger input is implemented as an opto coupler in series with a 1K resistor. When using a PT1000CL Pleora-Box in combination with a VDS Vosskühler CPP-1000 or CLA-1 module, the trigger input can be found on the Pleora-Box'es 12-pin Hirose connector (pin 10). On the Pleora-Box, the trigger input is implemented as a TTL-input (refered as TTL_IN0 in all Pleora manuals).
Note: When using a GIP-1000IR module, the timer mode will only have effect, if the module has been configured to “W=1” using a serial control terminal. In “W=0” mode (default), the camera's TREX (CC1) input will be directly connected to the trigger input. This configuration will always behave identically to “Timer Mode = off”.
Note: When using timer modes other than “off”, most cameras will require to enable the “Image On Demand” mode!

3.6.1 Timer Mode = Off

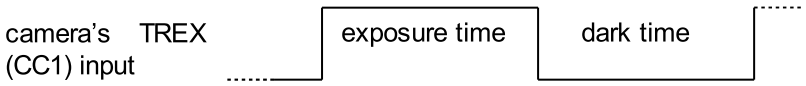
When this mode is activated, the trigger input is directly routed to the camera's TREX pin (respectively CC1 signal). The width of the trigger pulse directly determines the exposure time of the camera.



[2] VDS Vosskühler also uses the term “Interval Time“. (Interval Time = Exposure Time + Dark Time)

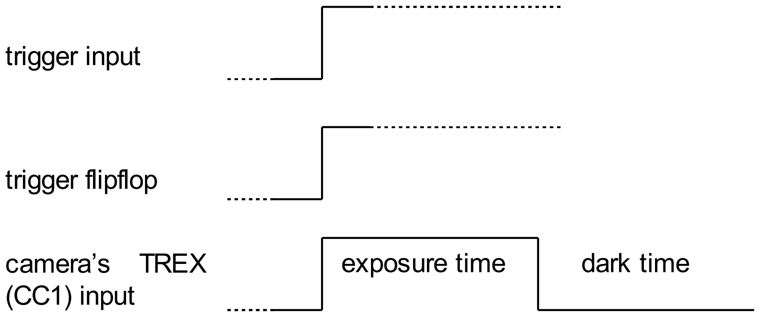
3.6.2 Timer Mode = Continuous

In this mode, a continuous exposure is started dependent on the “Exposure Time ms” and “Dark Time ms” parameter. The signal generation is independent on the trigger input.



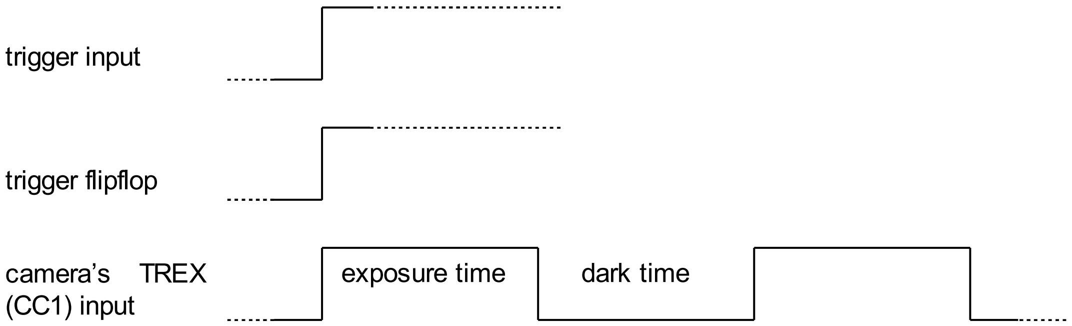
3.6.3 Timer Mode = Trigger Once

In this mode, a single exposure cycle of definite length is emitted upon a trigger event. The exposure time and the dark time are specified by the “Exposure Time ms” and “Dark Time ms” parameter. The trigger input sets the trigger flipflop which then starts the exposure cycle. No further exposure cycles will be emitted until the trigger flipflop is reset by the software. See parameter “Reset Trigger Flipflop”. Only when the trigger flipflop is not yet set, the circuit will be sensitive to new trigger events.



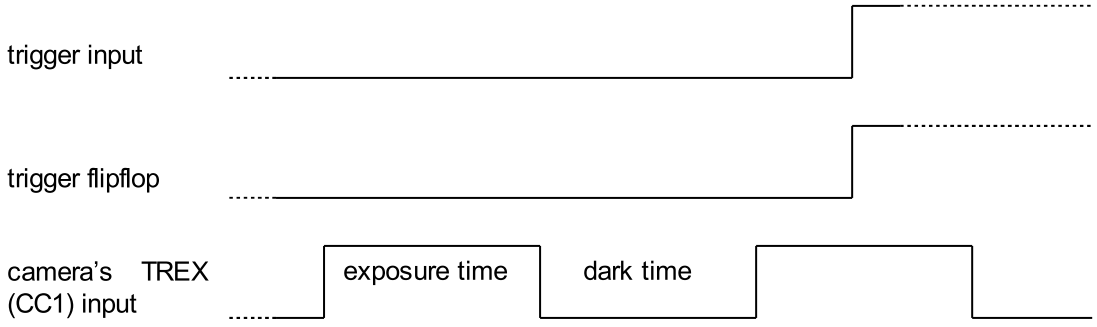
3.6.4 Timer Mode = Start By Trigger

In this mode, a continuous exposure is started upon a trigger event. The exposure time and the dark time are specified by the “Exposure Time ms” and “Dark Time ms” parameter. The trigger input sets the trigger flipflop which then starts the exposure series. Only when the trigger flipflop is not yet set, the circuit will be sensitive to a trigger event. The trigger flipflop must be reset by software . See parameter “Reset Trigger Flipflop”.



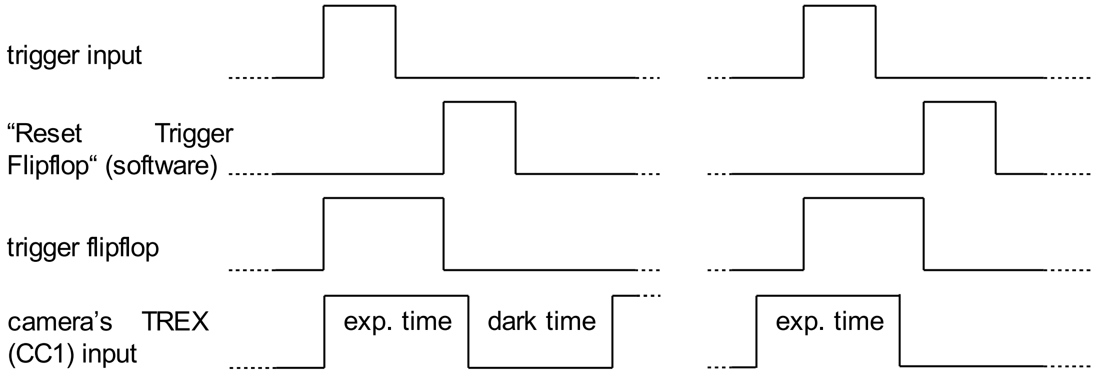
3.6.5 Timer Mode = Stop By Trigger

In this mode, a continuous exposure is started by software. The exposure time and the dark time are specified by the “Exposure Time ms” and “Dark Time ms” parameter. The trigger input sets the trigger flipflop which then synchronously stops the exposure series. Only when the trigger flipflop is not yet set, the circuit will be sensitive to a trigger event. The trigger flipflop must be reset by software . See parameter “Reset Trigger Flipflop”.



3.6.6 Timer Mode = Start Stop By Trigger

In this mode, a continuous exposure signal can be started and stopped by a trigger event. Before a new trigger event is accepted, the trigger flipflop must be reset by software.


trigger input
“Reset Trigger Flipflop“ (software)
trigger flipflop
camera’s TREX (CC1) input
exp. time
dark time
exp. time

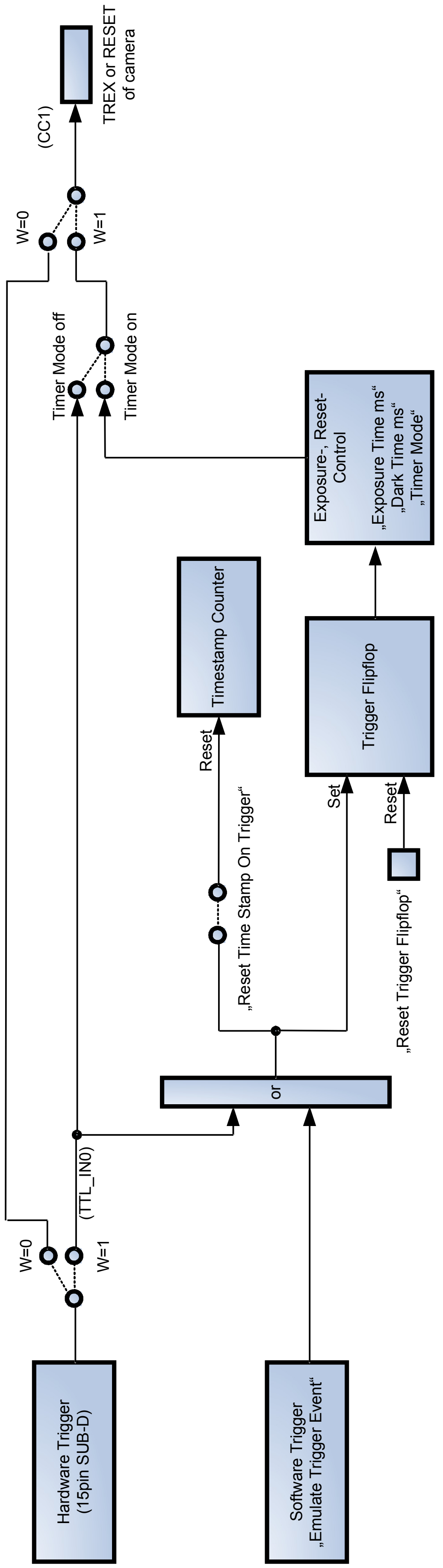


Figure 3 – The structure of the exposure control unit

3.7 Reset Trigger Flipflop/Emulate Trigger Event

These parameters are used to reset or explicitly set the trigger flipflop. Dependent on the timer mode, the exposure signal generation is changed upon the rising edge of the trigger flipflop's output. Before a trigger event can have effect, the trigger flipflop must be reset by software. The parameter "Reset Trigger Flipflop" can be used to archieve this. If it is desired to simulate a trigger event by software, the parameter "Emulate Trigger Event" can be used. These parameters can have tree states: "Off", "On" and "Broadcast".

State	Description
Off	Do not change the state of the trigger flipflop
On	Reset/Set the trigger flipflop
Broadcast	Reset/Set the trigger flipflop on all cameras present on the network (this can be used to synchronize multiple cameras

Table 3 – Trigger Flipflop modification

If both "Reset Trigger Flipflop" and "Emulate Trigger Event" parameters are set to "On" or "Broadcast", the trigger flipflop is first reset and then set again during applying parameter changes.

3.8 Reset Time Stamp On Trigger

If this parameter is set, the timestamp counter is reset upon a trigger event. This is useful for example, when using multiple synchronized cameras. The timestamp can then be used to assign images from different cameras to certain times.

3.9 Gain

Most cameras from VDS Vosskühler are equipped with the ability to amplify the camera image before digitalization. If this parameter is set, the amplification is active.

3.10 Image On Demand

Most cameras from VDS Vosskühler provide the ability to switch between continuous mode and image on demand mode. In continuous mode, the camera generates its own exposure timing and records images arbitrary. In image on demand mode, it is required to apply an external exposure signal. The timing control unit can be used to generate this timing. See chapter 3.3 for timer modes.

3.11 Video Mode

The video mode is a grouping of camera parameters that depend on each other, such as binning, pixel depth and resolution. The available entries depend highly on the camera model that is being used.

3.12 Dark Value Control

This parameter is currently only implemented on VDS Vosskühlers IRC-300 camera series. It is used to adjust the camera image's offset to a reference value. See the camera manual for further details.

3.13 Peltier Control

This parameter is currently only implemented on VDS Vosskühlers IRC-300 camera series. It is used to enable/disable a reference element that is required for the "Dark Value Control" feature. See the camera manual for further details.

4 Setting Parameters using C++

This chapter describes, how to modify the VDS Vosskühler specific parameters in your own software. It assumes, that you have included the Pleora iPort SDK in your software project and that you have access to the SDK's functionality. If you are not yet familiar with the Pleora iPort SDK, you are encouraged to have a look at the related Pleora documentation called “C++ SDK Reference Guide.chm”, “iPort.Reference.C++_SDK.chm” or similar. The documentation is usually installed into the “Documents”-directory relative to the Pleora SDK's installation base directory.

The usual approach to get access to the Camera Integration's DLL's functionality, is by using the CyCameraRegistry object. This object can be used to create an instance of a CyCameraInterface-object, that directly maps to the VDS Vosskühler specific features.

```
CyCameraRegistry   tRegistry;
CyResult           tResult;
CyCameraInterface* ptCameraInterface;
...

// configure the CyCameraRegistry object to the currently selected camera model.
// NOTE: m_tGrabber must be a connected grabber object.
if ((tRet = tRegistry.FindCamera(“CCD-1300B”)) == CY_RESULT_OK)
{
      // now create an instance of the camera interface from the
      // VDS Vosskuehler's "Camera Integration DLL".
      if (
            (tRet = tRegistry.CreateCamera(&ptCameraInterface, &m_tGrabber))
            == CY_RESULT_OK
         )
      {
            // camera interface creation successful
            ...
      }
      else
      {
            // could not create camera interface
            ...
      }
}
else
{
      // camera not found - is VDS Vosskuehler's integration DLL installed?
      ...
}
```

If you have successfully instantiated the CyCameraInterface-object, you are ready to set/get the parameters using the CyCameraInterface's „SetParameter“ and „GetParameter“ method. The valid parameter ID's are defined in the “CyVosskuehlerCamera.h” file that is part of this package. After installation, it can be found in the “Includes” directory of the Pleora base tree. This file must be included in your application. It defines the following parameter ID's:

```
#define CY_CAMERA_PARAM_VDS_EXP_TIME                      0x80000001
#define CY_CAMERA_PARAM_VDS_DARK_TIME                     0x80000002
#define CY_CAMERA_PARAM_VDS_GAIN                          0x80000003
#define CY_CAMERA_PARAM_VDS_IOD                           0x80000004
#define CY_CAMERA_PARAM_VDS_VIDEOMODE                     0x80000005
#define CY_CAMERA_PARAM_VDS_DVC                           0x80000006
#define CY_CAMERA_PARAM_VDS_PELTIER                       0x80000007
#define CY_CAMERA_PARAM_VDS_EXP_TIMER_MODE                0x80000008
#define CY_CAMERA_PARAM_VDS_RESET_TRIGGERFF               0x80000009
#define CY_CAMERA_PARAM_VDS_EMULATE_TRIGGER_EVENT         0x8000000A
#define CY_CAMERA_PARAM_VDS_RESET_TIMESTAMP_ON_TRIGGER    0x8000000B
#define CY_CAMERA_PARAM_VDS_TRIGGERFF_STATE               0x8000000C
#define CY_CAMERA_PARAM_VDS_CAMID                         0x8000000D
#define CY_CAMERA_PARAM_VDS_FTDURATION                    0x8000000E
#define CY_CAMERA_PARAM_VDS_MINDARKTIME                   0x8000000F
```

Note that not all parameters are defined for all cameras of VDS Vosskühler. To query the presense of a particular parameter, call the “HasParameter”-method of the CyCameraInterface-object. You also can do a complete parameter enumeration using the “GetParameterCount” and “GetIdentifier”-methods.

To set the exposure time of a “CCD-1300B” or similar camera to 100ms recording a new image every 300ms, you simply need the following code:

```
ptCameraInterface->SetParameter(CY_CAMERA_PARAM_VDS_EXP_TIME, 100.0);
ptCameraInterface->SetParameter(CY_CAMERA_PARAM_VDS_DARK_TIME, 200.0);
ptCameraInterface->SetParameter(CY_CAMERA_PARAM_VDS_IOD, 1);
ptCameraInterface->SetParameter
      (
            CY_CAMERA_PARAM_VDS_EXP_TIMER_MODE,
            (__int64) EVDS_TIMER_CONTINUOUS
      );

ptCameraInterface->UpdateToCamera();
```

For more detailed information on setting/getting/enumarating parameters, please have a look into the Pleora's SDK documentation.

5 Example applications

The Camera Integration DLL provides some special features to trigger or reset one or more CCD-, NIR-, or IRC-cameras.

5.1 Synchronizing 2 IRC-300GE cameras

The idea of synchronizing 2 or more IRC-300GE cameras is, to trigger (or reset) all cameras at the same time and also reset the internal timestamp of the Pleora engine. As a result you will get images that are recorded nearly at the same time and all with nearly the same timestamp. Of course, a hardware trigger will result in a more precise synchronization than a software trigger (via broadcast). In a test application with 2 cameras we measured a trigger delay between both cameras of less than 2ms with the software-broadcast method.

If you use the Coyote software for tests, then you should set the following parameters in both (all) cameras:

- Send "W=1" using the "Port Communication" to every camera.
- "Timer Mode = Trigger Once“
- "Reset Trigger Flipflop = Broadcast"
- "Reset Time Stamp On Trigger = true“

a) For a software trigger you can use "Emulate Trigger Event" as "Broadcast".
b) For a hardware trigger you have to activate the trigger-inputs (pin 10, 11 of the 15-pin SUB-D connector) of all cameras at the same time.

Note: The availability of the trigger feature depends on the hardware/firmware version of the GIP-1000 module.

6 Contact Information

If you need further information or support regarding this software, you can contact us via internet at www.vdsvossk.de or by telefone under +49 / (0)541 / 80084-0.